Image Segmentation in Twenty Questions

Christian Rupprecht[1,2] Loïc Peter[1] Nassir Navab[1,2]

[1]Technische Universität München, Munich, Germany
[2]Johns Hopkins University, Baltimore MD, USA
{christian.rupprecht, peter, navab}@in.tum.de

Abstract

Consider the following scenario between a human user and the computer. Given an image, the user thinks of an object to be segmented within this picture, but is only allowed to provide binary inputs to the computer (yes or no). In these conditions, can the computer guess this hidden segmentation by asking well-chosen questions to the user? We introduce a strategy for the computer to increase the accuracy of its guess in a minimal number of questions. At each turn, the current belief about the answer is encoded in a Bayesian fashion via a probability distribution over the set of all possible segmentations. To efficiently handle this huge space, the distribution is approximated by sampling representative segmentations using an adapted version of the Metropolis-Hastings algorithm, whose proposal moves build on a geodesic distance transform segmentation method. Following a dichotomic search, the question halving the weighted set of samples is finally picked, and the provided answer is used to update the belief for the upcoming rounds. The performance of this strategy is assessed on three publicly available datasets with diverse visual properties. Our approach shows to be a tractable and very adaptive solution to this problem.

1. Introduction

Twenty Questions is a classical two-player game involving a questioner and an oracle. Before the game starts, the oracle thinks about something (*e.g.* an object, an animal or a character) which we will call the answer, and the questioner is allowed to ask a series of binary questions to guess what the answer is. While the game originally involves a human in the role of the oracle and another human as player, the development of artificial intelligence techniques led to softwares where the computer tries to guess what the human user has in mind [1, 2]. Going in the same direction, we consider in this work the case of a computer playing the Twenty Questions game with a human user in the role of the oracle, where the expected answer is a *binary segmentation* of a given image. We introduce a strategy for the computer to provide a guess as accurate as possible in a limited number of questions. Alternatively, the proposed approach can also be seen as an interactive segmentation task. The crucial characteristic of our scenario is that the user interaction with the machine is restricted to binary inputs (*yes*/*no*) instead of the usual scribbles or bounding boxes [10]. This setting is common to provide relevant feedback for interactive image retrieval [19, 24, 26] but was never considered in the context of segmentation. Moreover, if combined with *e.g.* a voice recognition system, our method can eventually provide a hands-free segmentation technique, for which an alternative solution was proposed by Sadeghi *et al.* [21] who used an eye tracker for seed placement. This has potential applications like segmentation of medical data in sterilized operating room environments, where physical interactions with objects have to be avoided.

Our method can be summarized as follows. At each turn, the question to be posed consists in asking whether a well-chosen pixel is inside the hidden segmentation or not. The choice of the location to ask about is made as to maximize the information brought by the answer. More precisely, we define for any possible segmentation a probability to have been picked by the oracle. Each segmentation score is based on two aspects: its intrinsic quality in terms of image partitioning (*i.e.* the homogeneity of the segmented object), and its compatibility with the answers already collected. Hence, this defines a probability distribution over the set of possible segmentations $\mathcal{S}$. In theory, the ideal question choice would follow a divide and conquer approach and halve $\mathcal{S}$ in two parts of approximately equal probabilities. However, the huge size of $\mathcal{S}$ excludes the exhaustive computation of the aforementioned distribution, which we approximate instead by sampling a series of segmentations proportionally to their probability of having been chosen by the oracle. This sampling is performed via a Markov Chain Monte Carlo (MCMC) framework, which builds on the Metropolis-Hastings algorithm [18] and consists in ex-

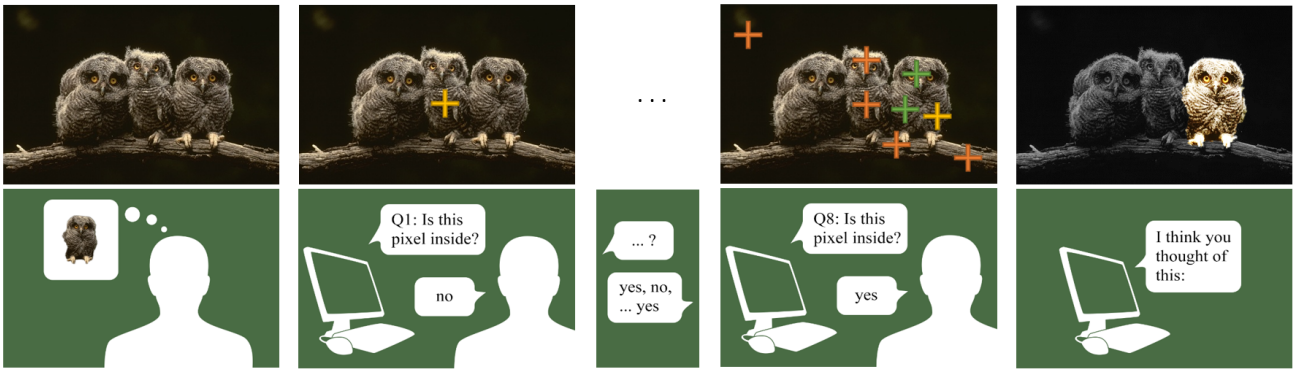

Figure 1: **Overview of the Twenty Questions segmentation scenario.** Given an image, the object to segment is secretly chosen by the human user. At every step, the computer asks whether a certain pixel is located inside the desired segmentation. After a predefined number of questions, it returns its guess about the answer.

ploring $\mathcal{S}$ while spending more time in areas of high probability. The most informative question can be computed from the drawn samples in a tractable way, and the knowledge brought by the answer is included in the probabilistic model for the future rounds. We evaluate our method on three publicly available segmentation datasets with various visual properties and compare it with several baselines. The experiments demonstrate that our approach is a promising solution to the problem of interactive object segmentation with binary inputs only. The question selection strategy is fast enough (around 1 second) to be incorporated in a software in practice and does not rely on any offline training step, which also makes the framework overall very adaptive to different types of images.

Related Work In this paragraph, we briefly review the connections of our work with existing methods. The Twenty Questions setting has already been mentioned in the computer vision community through the work of Branson *et al*. [5] and its extension by Wah *et al*. [28] for interactive fine-grained image classification. These works consider the case where both the human user and the machine ignore the image label, but (i) the machine knows which questions are important to ask to find out the answer and (ii) the human is able to answer these questions which are based on the visual aspect of the scene. Hence, combining the expertise of the computer with the visual abilities of the human allows to find collaboratively the hidden image label. Beyond the differences in terms of task (image classification *vs* segmentation), this setting is fundamentally different from ours, where the object to be segmented is perfectly known by the human user and has to be guessed by the computer.

Interactive segmentation techniques usually rely on seeds [4, 10, 20] or bounding boxes [11, 16] that are manually placed by a human user in or around the object of interest. Closer to our work, a few approaches [3, 8, 13, 14, 23] keep the human user in the loop and suggest the most informative areas to label next, in an active learning fashion. An important aspect of our approach is the intrinsic ambiguity of the image parsing task, as one cannot anticipate the semantic level of the segmentation picked by the oracle. In this direction, Tu and Zhu [27] introduced a data-driven MCMC framework based on active contours able to generate several parsings of a same image. Recent alternatives identify a set of candidate relevant objects in a scene [6, 7, 12, 15] by learning plausible object appearances or shapes. In our case, no offline training is performed so that the method can be applied to any kind of image or ground truth.

2. Methods

2.1. Problem Statement and Notations

An image $\mathcal{I}$ is defined over a lattice $\Omega = \{1, \dots, h\} \times \{1, \dots, w\}$ where h and w respectively denote the height and width of $\mathcal{I}$. We define a binary segmentation of the image $\mathcal{I}$ as a function $\mathbf{s} : \Omega \to \{0, 1\}$ indicating whether the pixel (x, y) of the image is inside ($\mathbf{s}(x, y) = 1$) or outside ($\mathbf{s}(x, y) = 0$) the segmented area. Depending on the context, $\mathbf{s}$ can also be seen as a vector $\mathbf{s} \in \mathcal{S} = \{0, 1\}^{|\Omega|}$.

Our problem can be formalized as follows. Initially, given a fixed image $\mathcal{I}$, the oracle (*i.e*. the human user) decides on a segmentation $\hat{\mathbf{s}} \in \mathcal{S}$ that has to be found by the computer. To do so, the computer is going to ask to the oracle a series of binary questions of the form $Q(\mathbf{p})$: "*Is the displayed location* $\boldsymbol{p}$ *inside the object you want to segment?*". Choosing the best question to ask amounts to finding the most informative location $\mathbf{p}$. In return, the answer directly provides the true label $l(\mathbf{p}) \in \{0, 1\}$ at this location.

After k questions have been posed, the collected answers provide two reliable sets Σ_-^k and Σ_+^k of background and foreground seeds respectively, with $\left|\Sigma_-^k\right| + \left|\Sigma_+^k\right| = k$. We

also denote $\Sigma^k = \Sigma^k_- \cup \Sigma^k_+$ the set of reliable seeds collected. This knowledge is encoded through a Bayesian posterior probability $p(\mathbf{s} = \hat{\mathbf{s}}|\Sigma^k)$ over the set of segmentations $\mathcal{S}$ stating how likely it is that the segmentation $\mathbf{s}$ has been initially picked by the user given the known seeds revealed by the answers already collected. We will denote this probability $p(\mathbf{s}|\Sigma^k)$ in the rest of the paper to make clear that this probability is seen as a function of $\mathbf{s}$ and as a probability distribution over $\mathcal{S}$. If this posterior could be computed for every possible segmentation in $\mathcal{S}$, an optimal divide and conquer strategy would halve at each turn the set of possible segmentations into two subsets of probability 0.5 each. However, the set $\mathcal{S}$ has an extremely large size, with theoretically $2^{|\Omega|}$ possibilities, which excludes the exhaustive computation of $p(\mathbf{s}|\Sigma^k)$ over the whole set $\mathcal{S}$. To overcome this, we propose at each iteration k to approximate the posterior $p(\mathbf{s}|\Sigma^k)$ by a series of samples $\mathbf{s}^k_1, \ldots, \mathbf{s}^k_N \in \mathcal{S}$ drawn according to a Markov Chain Monte Carlo (MCMC) scheme, which is described in detail in Sec. 2.2. After these N samples have been drawn, we select the most informative question based on these samples by following the question selection method exposed in Sec. 2.3. These two sampling and question selection steps are then iterated until a predefined amount of allowed questions is reached.

Input : Image $\mathcal{I}$, number of allowed questions K
$\Sigma^0_- \leftarrow \emptyset$; $\Sigma^0_+ \leftarrow \emptyset$;
for $k \leftarrow 0$ **to** $K-1$ **do**
  Sample $\mathbf{s}^k_1, \ldots, \mathbf{s}^k_N$ from $p(.|\Sigma^k)$;
  Find most informative location $\mathbf{p}_k$ w.r.t $\mathbf{s}^k_1, \ldots, \mathbf{s}^k_N$;
  Ask question $Q(\mathbf{p}_k)$ and receive true label $l(\mathbf{p}_k)$;
  if $l(\boldsymbol{p}_k) = 0$ **then**
    $\Sigma^{k+1}_- \leftarrow \Sigma^k_- \cup \{\mathbf{p}_k\}$;
    $\Sigma^{k+1}_+ \leftarrow \Sigma^k_+$;
  else
    $\Sigma^{k+1}_- \leftarrow \Sigma^k_-$;
    $\Sigma^{k+1}_+ \leftarrow \Sigma^k_+ \cup \{\mathbf{p}_k\}$;
  end
end
$\mathcal{G} \leftarrow$ GDT-segmentation with seeds Σ^K;
Output: Guess $\mathcal{G}$ of the oracle segmentation

Algorithm 1: Overview of our method. At each step, the question selection strategy consists in sampling segmentations according to their probability of having been picked by the oracle, and asking the question related to the most informative location with respect to these samples. This location together with the provided label form a seed which is added to either Σ^0_- or Σ^0_+ depending on its label.

2.2. Sampling Likely Segmentations with MCMC

In this section, we introduce our procedure for sampling representative segmentations from the posterior probability distribution $p(.|\Sigma^k)$ conditioned on the current knowledge. Following the classical Metropolis-Hastings algorithm [18] and an original idea from Tu and Zhu [27] introduced in the context of image parsing into multiple regions, we define our sampling procedure as a Markov chain over the space of segmentations with transition probabilities defined as follows. Given a current segmentation $\mathbf{s}$, a new segmentation candidate $\mathbf{s}'$ is suggested according to a *proposal distribution* $q(\mathbf{s}'|\Sigma^k, \mathbf{s})$ to be the new state of the Markov chain. This move is accepted with probability $\min(1, \alpha)$ with

$$\alpha = \frac{p(\mathbf{s}'|\Sigma^k) q(\mathbf{s}|\Sigma^k, \mathbf{s}')}{p(\mathbf{s}|\Sigma^k) q(\mathbf{s}'|\Sigma^k, \mathbf{s})}, \tag{1}$$

where $p(.|\Sigma^k)$ denotes the *posterior probability distribution* according to which we want to draw samples, *i.e.* the probability for a segmentation to have been picked by the user given the currently known seeds Σ^k. Starting from an initial segmentation $\mathbf{s}^k_0$, a succession of segmentations is generated and the $\mathbf{s}^k_i, 1 \leq i \leq N$ are selected as samples at a fixed rate during this exploration process. Additionally, a burn-in step is performed before starting collecting samples so that the dependency on the initial state $\mathbf{s}^k_0$ is strongly reduced. To complete the description of the segmentation sampling method, one has to carefully define the posterior distribution and the proposal distribution. Our design for these distributions will be described in Sec. 2.2.2 and Sec. 2.2.3 respectively. They both build on a parametrization of the set of segmentations $\mathcal{S}$ which we are now going to introduce.

2.2.1 State-Space Parametrization

To facilitate the design of efficient proposal distributions in Eq. 1, we need a way to deform a segmentation into another. In their data-driven MCMC framework for image parsing, Tu and Zhu proposed to deform active contours [27]. In our case, the MCMC paradigm takes place between two questions in the context of a human/machine interaction. Hence, it is essential to keep the time between two questions as small as possible. For this reason, we introduce an alternative view of our segmentation space based on geodesic distance transforms (GDT) that recently proved to be extremely efficient in the context of object proposals [15].

We define $\mathcal{X} = \{1, \ldots, C\} \times [0, B] \times \mathcal{P}(\Omega)^2$ as our state space and consider the function $\phi^{GDT} : \mathcal{X} \rightarrow \mathcal{S}$ that associates to a vector $\mathbf{x} = (c, \beta, \Sigma^+, \Sigma^-) \in \mathcal{X}$ the segmentation $\phi^{GDT}(\mathbf{x}) \in \mathcal{S}$ obtained by computing the geodesic distance transform on: the c^{th} image channel, blurred with a Gaussian of standard deviation β, with the sets of positive and negative seeds Σ^+ and Σ^- respectively. C denotes the number of color channels in the image, and B the maximally

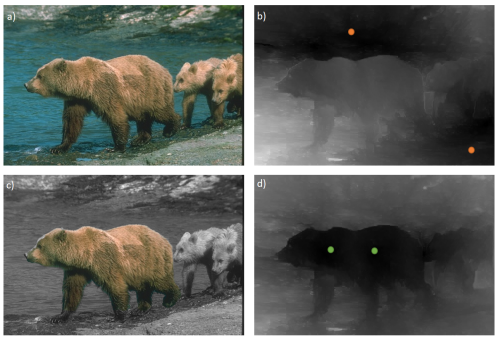

Figure 2: **Segmentation example using geodesic distance transforms.** a) Image from the Berkeley 300 dataset [17]. b) + d) Approximated geodesic distance to outside (orange) and inside (green) seeds respectively. c) Final segmentation.

allowed standard deviation. The GDT is obtained by computing the shortest distance of every pixel to the set of seed pixels. Usually, the distance between two neighboring pixels (*i.e.* the edge weights on the image graph) is defined as a mixture of the Euclidean distance and the gradient between these two points. To release the dependence on the seed placement within the object of interest, we use the squared intensity difference only. This is particularly important because there is no guarantee that seeds will be in the center of objects since the user is no longer placing the seeds manually. To generate the final segmentation, each pixel receives the label of its closest seed. The coefficient $\beta \in [0, B]$ is introduced to control the sensitivity to edges in the image and implicitly generates an image pyramid during the MCMC process (see Sec. 2.2.3). Note that any other seed-based interactive segmentation algorithm could be included at this stage of the framework instead. Our main motivation behind the choice of geodesic distance transforms is the fact that they can be approximated in linear time [25] and are hence very fast to compute. The GDT-based segmentation procedure is further illustrated in Fig. 2.

The Markov chain used for our segmentation sampling is going to act at the state space level, *i.e.* on the GDT-based segmentation parameters and seeds. Eq. 1 becomes

$$\alpha = \frac{p(\phi^{GDT}(\mathbf{x}')|\Sigma^k)q(\mathbf{x}|\Sigma^k, \mathbf{x}')}{p(\phi^{GDT}(\mathbf{x})|\Sigma^k)q(\mathbf{x}'|\Sigma^k, \mathbf{x})}. \quad (2)$$

The two next subsections are going to expose our design for $p(\phi^{GDT}(\mathbf{x})|\Sigma^k)$ and $q(\mathbf{x}'|\Sigma^k, \mathbf{x})$ respectively.

2.2.2 Posterior Probability

The probability $p(\mathbf{s}|\Sigma^k)$ states the probability of the segmentation to have been initially picked by the user given the set of k seeds Σ^k already revealed by the answers to the k first questions. An important characteristic of the Metropolis-Hastings acceptance probability (Eq. 2) is the fact that only the ratio of the probabilities $p(\phi^{GDT}(\mathbf{x}')|\Sigma^k)$ and $p(\phi^{GDT}(\mathbf{x})|\Sigma^k)$ appears. Hence, the normalization factor of this probability distribution does not play any role and we can design this distribution without taking it into consideration. To define this probabilistic term, we propose to distinguish two cases depending on whether there is at least one background seed and one foreground seed in Σ^k.

Case 1: $\Sigma^k_+ = \emptyset$ or $\Sigma^k_- = \emptyset$ This case, typically occurring during the first questions, corresponds to the absence of seeds for at least one of the two labels. Note that it occurs a least for the two first turns, where our knowledge consists of respectively 0 and 1 seed. We define the segmentation probability as

$$p(\mathbf{s}|\Sigma^k) \propto \frac{1}{1 + Var(\{\mathcal{I}(\mathbf{p}), \mathbf{p} \in \mathbf{s}^{-1}(1)\})}, \quad (3)$$

where $Var(\{\mathcal{I}(\mathbf{p}), \mathbf{p} \in \mathbf{s}^{-1}(1)\})$ denotes the variance of the image values over the set of foreground locations defined by the segmentation $\mathbf{s}$. This variance is summed over all color channels. Intuitively, we encourage segmentations which delineate homogeneous regions as foreground.

Case 2: $\Sigma^k_+ \neq \emptyset$ and $\Sigma^k_- \neq \emptyset$ Once at least one seed inside and outside the object is known, we can use the GDT segmentation algorithm to build a more accurate estimate of the visual properties of the background and foreground regions. For this, we perform the GDT segmentation based on the known seeds on each color channel and compute background and foreground intensity histograms H^k_- and H^k_+ aggregated over the color channels. While building the histograms, each intensity value is weighted by its inverse geodesic distance to encode the fact that the confidence decreases with increasing (geodesic) distance to the seeds. To assign a score to a new segmentation $\mathbf{s}$, we compute similarly the background and foreground histograms of $\mathbf{s}$ denoted $H_-(\mathbf{s})$ and $H_+(\mathbf{s})$ and measure their mismatch to the current estimates H^k_- and H^k_+ via a chi-squared distance:

$$p(\mathbf{s}|\Sigma^k) \propto \frac{1}{1 + \frac{1}{2}\sum_{\delta \in \{-,+\}} \chi(H^k_\delta, H_\delta(\mathbf{s}))}. \quad (4)$$

2.2.3 Proposal Distribution

Our segmentations are generated via a set of parameters $\mathbf{x} = (c, \beta, \Sigma^+, \Sigma^-) \in \mathcal{X}$ sent as input to the GDT segmentation algorithm. The main advantage of this representation is that the state space $\mathcal{X}$ gives a more natural way to move from a state to another and facilitates the design of the proposal distribution $q(.|\Sigma^k, \mathbf{x})$. In practice, we maintain two sets of seeds Σ^+ and Σ^- which contain both the already known and hence reliable seeds included in Σ^k_+ and Σ^k_-

Figure 3: **Over-segmentation of the image for the question selection step.** The over-segmentation is obtained by intersecting all $\mathbf{s}_i^k, 1 \leq i \leq N$. The most uncertain region R_j^* is shown in green and the question pixel $\mathbf{p}_k$ (shown in orange) is chosen as the most interior point of R_j^*.

(*fixed* seeds) and some other seeds created by the MCMC process exclusively (*mobile* seeds). From a given state $\mathbf{x}$, a state $\mathbf{x}'$ is suggested by drawing uniformly and performing one of the 5 following moves:

1. *Changing image channel*: The image channel c is redrawn uniformly.

2. *Changing* β: The smoothing parameter β is redrawn uniformly.

3. *Adding a seed*: A mobile seed is added at a random (non seed) location and randomly added to Σ^+ or Σ^-.

4. *Removing a seed*: A mobile seed is removed from either Σ^+ or Σ^- (if this does not leave this set empty).

5. *Moving a seed*: A mobile seed is moved spatially to a (non seed) pixel according to a normal distribution.

which entirely defines the proposal distribution $q(.|\Sigma^k, \mathbf{x})$. To avoid that the number of seeds diverges, we balance the probabilities of picking the moves 3. and 4. such that forward and backward moves are equally likely.

2.3. Question Selection

After k questions have been asked and answered ($k \geq 0$), the method described in Sec. 2.2 draws N segmentations $\mathbf{s}_i^k, 1 \leq i \leq N$ that approximate the probability distribution $p(\mathbf{s}|\Sigma^k)$ that the segmentation $\mathbf{s}$ is the answer awaited by the oracle given the current knowledge (encoded by Σ^k). From these samples, we have to decide on the optimal question to ask to the user. To do so, we first perform an over-segmentation of the image by intersecting all $\mathbf{s}_i^k$ (Fig. 3). This provides a partition of the image into regions $(R_j)_{1 \leq j \leq \rho}$ according to the samples $\mathbf{s}_i^k$. Since the $\mathbf{s}_i^k$ are the only available information, the information carried by each pixel is constant over an individual segment. The

Figure 4: **Representative images from the datasets.** From left to right: (1) Berkeley Segmentation Dataset, (2) Stanford Background Dataset, (3) IBSR Brain Dataset. Note the diversity in terms of visual content (natural scenes (1,2) *vs* medical data (3)) and labels (unsupervised parsing (1) *vs* semantic labeling (2,3)).

probability for a region R_j to belong to the object of interest $p(R_j \subset \hat{\mathbf{s}})$ can be approximated from our samples as

$$p(R_j \subset \hat{\mathbf{s}}) = \frac{1}{\sum_{i=1}^N p(\mathbf{s}_i^k|\Sigma^k)} \sum_{i|R_j \in \mathcal{S}_i} p(\mathbf{s}_i^k|\Sigma^k). \quad (5)$$

The most uncertain segment is the one for which this probability is the closest to 0.5. However, each segment has to be weighted by its size to account for the fact that each pixel will equally benefit from the information carried by the segment it belongs to. Thus we select the segment R_j^* maximizing $|R_j|(1 - |1 - 2p(R_j \subset \hat{\mathbf{s}})|)$ as the most informative. At this stage, we could indifferently choose any location $\mathbf{p}_k \in R_j^*$ to define the selected question $\mathcal{Q}(\mathbf{p}_k)$. In practice, we choose for $\mathbf{p}_k$ the most interior point of R_j^* (Fig. 3). Indeed, this results in questions that are in general easier to answer visually as the most interior point typically lies far from the edges within the image.

3. Experiments

3.1. Experimental Setup

We describe the parameter settings of our method, that are the same for the three datasets, in more detail. We use the CIELab color space and the maximum blur level β, *i.e.* the standard deviation of the Gaussian smoothing applied during the MCMC process (Sec. 2.2.1) is set to $B = 6$ pixels. The MCMC burn-in step, whose goal is to gain independence from the starting state $\mathbf{s}_0^k$, consists of 1000 iterations. 250 Markov chain moves are performed between two samples drawings to ensure a strong decorrelation between two consecutive samples. The total number of sampled segmentations at each turn is set to $N = 32$. It was empirically found to be sufficient to produce enough variety between the segmentations such that the over-segmentation does not become too fine grained without exceeding one second between two questions. The main bottleneck is the computation of the approximate GDT, required at each MCMC

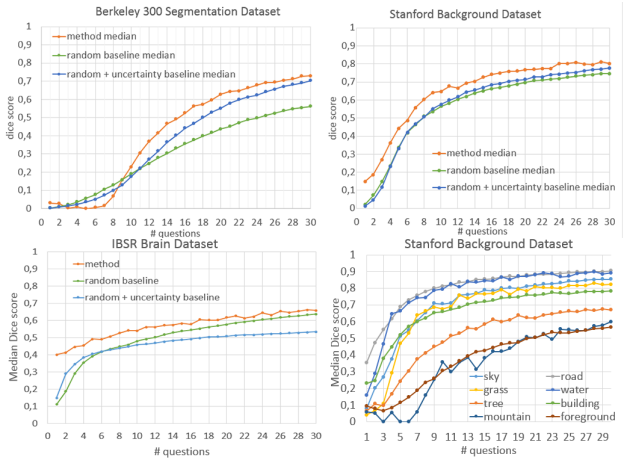

Figure 5: **Evolution of the median Dice scores with the number of questions.** The performance is shown on the three datasets: Berkeley Segmentation Dataset 300 [17], Stanford Background Dataset [9] and IBSR Brain Dataset. We also show, on the Stanford background dataset, the individual performance for each class label. The small amount of mountain labels in the dataset explains the noise on the corresponding curve.

step, which is linear in the number of pixels. Thus, the total complexity is $\mathcal{O}(NKwh)$. The Markov chain runs on a downsampled version of the image to speed up computation between the samples, but the drawn segmentations every 250 steps remain computed on the full image to ensure a fine grained segmentation. The method is evaluated for up to $K = 30$ questions. In our experiments, the human user is simulated by the ground truth segmentation, which allows an extensive evaluation over a large set of images. All experiments were performed on an Intel i7-4820K 3.7GHz CPU. The algorithm runs four independent Markov chains in parallel.

3.2. Results

To measure the quality of a segmentation, we use the Sørenson index or Dice score [22], which measures the overlap between the segmentation and the ground truth. We compare our method against two intuitive baselines. The first chooses each question by drawing randomly the corresponding pixel location. Our second baseline is an improvement of the first one: as soon as seeds have been found inside and outside the hidden segmentation, GDT-based segmentation is performed on these seeds. The algorithm uses the resulting pixelwise confidence to select the question within the area where the label uncertainty is maximal. In other words, this method can be seen as an iterative refinement of the segmentation border. The experimental evaluation was performed on three different datasets.

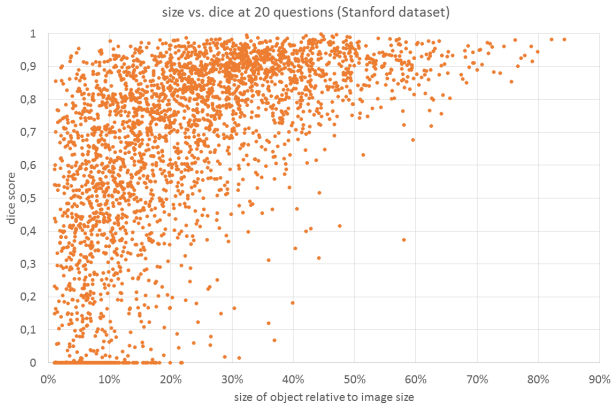

Figure 6: **Relationship between size of the hidden segmentation and performance.** Shown is a scatter plot of relative object size (x-axis) *vs* dice score (y-axis) after 20 questions. Large areas are easier to guess than small ones.

Berkeley Segmentation Dataset 300 The Berkeley Segmentation Dataset 300 [17] is a set of 300 natural images. For each of them, several ground truth human annotations defining a parsing of the image into several regions were collected. This dataset illustrates very well the complexity of finding the area the oracle thinks of, since the parsings are typically different from an annotator to another. The average number of regions in each individual ground truth is 20.3. To evaluate our method on this dataset, each region included in one of the ground truth parsings is selected as oracle segmentation. Figure 5 displays the median Dice score over all runs when the number of allowed questions increases, for both our method and the aforementioned baselines. The curves in Fig. 5 start with a plateau at low performance. This corresponds to the stage where no reliable foreground seed has been found yet, and hence the sampled segmentations are very noisy since they are only based on the mobile seeds from the Markov chain. Fig. 5 shows that the median number of questions required to find a reliable seed is approximately 7. This is interesting since the dataset contains 20 non-overlapping regions per image in average. Hence, even if this pool of 20 possible ground truth segmentations was known in advance, a random seed placement would still take about 10 questions to find a foreground seed. This demonstrates that our question selection procedure identifies regions more efficiently.

Stanford Background Dataset The second dataset used for evaluation is the Stanford Background Dataset [9] which is a diverse collection of 715 natural images taken from different other datasets. In this case, images have been labeled according to eight different semantic categories: sky, road, grass, mountain, water, tree, building and foreground object. This strongly differs from the Berkeley Segmentation

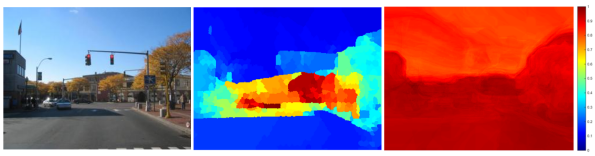

Figure 7: **Comparison between our MCMC-based proposals and geodesic object proposals.** The heat maps correspond to the frequency of occurrence of each pixel in the set of sampled segmentations, for the geodesic object proposal method [15] (middle) and our MCMC sampling scheme (right). While the former focuses more specifically on some areas, ours explores more the segmentation space.

Dataset where images were parsed into arbitratry regions which was intrinsically more ambiguous. The results are shown on Fig. 5. To give further insights, we also display the performance for each individual label (Fig. 5) and the relationship between Dice score after 20 questions and size of the target segmentation (Fig. 6). Bigger segmentations (respectively segmentations of a typically predominant class) appear to be easier to guess than smaller ones. This correlation between object size and segmentation quality is rather natural and directly comes from the type of questions that are asked. This point is discussed in more details in Sec. 3.3.

IBSR MR Brain Data To assess further the generality of our method, we evaluated it on the medical IBSR MR brain datasets composed of 18 magnetic resonance brain scans together with labels of 4 different brain structures. We decomposed the scans into $2D$ slices to have a larger number of runs for our evaluation. The data and their manual segmentations were provided by the Center for Morphometric Analysis at Massachusetts General Hospital and are available at `http://www.cma.mgh.harvard.edu/ibsr/`. Labeling the brain structures is a very complex task which requires a high expertise. The results of our method on the IBSR dataset can be seen in Fig. 5. As expected, the absolute performance is lower than on the two other datasets which had a richer visual content, but a significant improvement after 20 questions remains achieved.

Comparison with geodesic object proposals Finally, we compared our MCMC-based sampling of segmentations described in Sec. 2.2 with the recent geodesic object proposals (GOP) introduced by Krähenbühl and Koltun [15], whose goal is to suggest objects of interest in an image. By using the available (pre-trained) code for GOP, we sample for each image both N object proposals with GOP and N segmentations with our sampling method, where N is automatically inferred by the GOP algorithm. We report the Jaccard index of the sample matching the ground truth best (ABO score [15]). The GOP obtain an average score of 66.0 on the Stanford Background Dataset and 43.1 on the ISBR, while

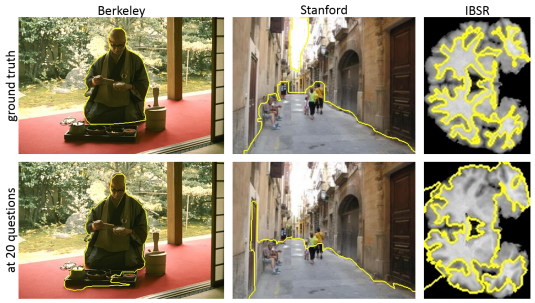

Figure 8: **Qualitative results.** Examples of oracle segmentation and the corresponding segmentation guess after 20 questions. From left to right: Berkeley Segmentation Dataset, Stanford Background Dataset, IBSR Brain Dataset.

our method obtains respectively 59.6 and 51.0. GOP appear to have an advantage on the Stanford Background Dataset, presumably because they rely on a seed placement technique which was learned precisely on outdoor scenes. On the contrary, our method does not rely on any offline training step, *i.e.* makes no assumption on the image content, and still performs better on brain images which illustrates the flexibility of our framework. Figure 7 shows another interesting difference between the two proposal frameworks. Without prior knowledge on the current image, the segmentations proposed by GOP are more redundant than ours, which is a good property to propose the likeliest object but suits less the context of an interactive guessing scenario.

Dataset	Method	10Q	20Q	30Q
Berkeley	Random	30.9 / 19.0	41.7 / 43.7	48.9 / 56.2
	Uncertainty	32.5 / 17.5	46.9 / 55.0	55.9 / 70.3
	Ours	34.7 / 23.8	48.8 / 62.0	56.1 / 73.2
Stanford	Random	49.8 / 56.5	60.2 / 69.8	65.6 / 74.7
	Uncertainty	50.6 / 57.6	61.9 / 71.4	67.9 / 77.7
	Ours	52.6 / 63.9	63.9 / 75.8	67.9 / 79.8
IBSR	Random	43.1 / 47.8	52.0 / 57.6	58.1 / 63.6
	Uncertainty	41.7 / 45.8	46.2 / 50.7	49.2 / 53.3
	Ours	51.5 / 53.7	58.4 / 60.1	62.9 / 64.7

Table 1: **Quantitative Results.** Mean and median Dice score after 10, 20 and 30 questions on the three datasets for our method and the two baselines described in Sec. 3.2

Inaccurate Answers To explore the robustness of the proposed method with respect to inaccurate answers to the questions, we tried an alternative experimental setting on the SBD dataset such that, when a question is posed within 3 pixels of the border of the desired object, the answer is randomly decided (50% yes 50% no). This happens about 20% of the time, mainly on the later questions. In average, 1.2 additional questions are needed to achieve the same Dice score as with perfect answers. Hence, the proposed method is reasonably robust to wrong answers.

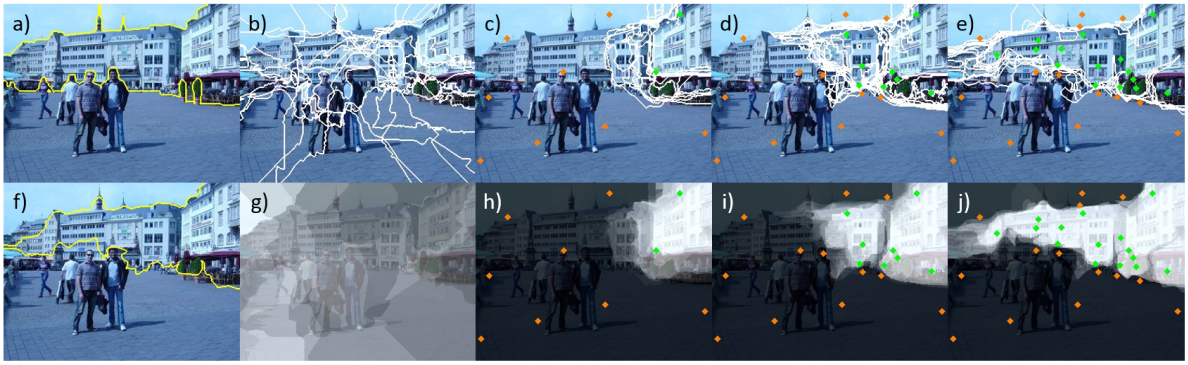

Figure 9: **Evolution of the segmentation belief with the number of answers collected.** a) shows the original image and the segmentation chosen by the oracle. Below, f) shows the final segmentation proposed by the computer after 30 questions. The images on the right show the evolution of the segmentation belief with the number of seeds collected (b: 0, c: 10, d: 20, e: 30). Both the intersection of all sampled segmentations (*i.e.* the over-segmentation described in 2.3 and shown in Fig. 3) and a heat map of the pixel likelihood are shown, respectively on the top and bottom row. The heat maps are generated by computing, for each pixel, the proportion of sampled segmentations that contained it (similarly to Eq. 5). The positive (resp. negative) seeds collected from the answers are shown in green (resp. orange). Note how the questions are progressively asked towards the left where the uncertainty is the highest, so that the whole series of buildings can be eventually segmented.

3.3. Discussion

We start this discussion with an example to illustrate further the behavior of the method. Figure 9 shows a typical instance of the scenario. The oracle thinks of an object, which is in this case the row of buildings in the background. It is a challenging target object for a segmentation algorithm due to the complexity of its structure and the fact that it corresponds to a rather high-level semantic. The computer generates a first set of possible segmentations. Since no seeds are available yet, uniform regions like sky or pavement are favored at this stage. The first question is asked at the location bringing the most information according to the drawn samples. As answers from the oracle are progressively collected, we can see that the space of sampled segmentations changes drastically. The algorithm suggests questions at boundary locations, and expands step by step the initial guess towards the left. A good estimate of the oracle segmentation is eventually provided after 30 questions.

The results demonstrate overall that our method has more difficulties to segment small objects, and in particular to find a first foreground seed in these cases. While it seems relatively intuitive that it takes longer to find small areas than big ones, the nature of the questions we ask is partially responsible for this as well. We investigated this by changing the kind of question asked. For instance, a more efficient solution to find small objects would be to show a highlighted area to the user and ask "*Is the target segmentation fully inside the shown area?*". If the answer is *yes*, all unhighlighted pixels could be added to the outside seeds and the search space would be greatly reduced. However, a *no* would provide almost no information since it would only state that at least one (unknown) pixel of the object is outside the shown area. Hence, this kind of questions suffers from the opposite problem: they are efficient to find small objects but poor at finding large ones, and have the additional drawback that a negative answer is practically very difficult to leverage. This is mainly what motivated us to choose the location-based question type, but we believe that a combination of both types of questions would be ideally suited to find both small and large objects in the image.

4. Conclusion

We introduced a method able to guess a segmentation in an arbitrary image by asking simple binary questions to the user. The questions are computed by approximating a posterior probability distribution over the set of segmentations with a MCMC sampling algorithm building on geodesic distance transform segmentation. Our method shows to be tractable for practical use with a waiting time between two questions of less than one second. No assumption about the type of the given image is necessary as the framework does not rely on any offline training step. The experimental evaluation performed on three very different datasets demonstrates that the approach provides an overall efficient solution to this problem. Our future directions of work include investigating the feasibility of mixing several question types and automatically learning the most relevant ones for the hidden segmentation. Since answering binary questions is plausibly more attractive than manual annotation, we are also considering using this method to generate a lot of training data for seed placements in a crowdsourcing fashion.

Acknowledgements With the support of the Technische Universität München - Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under grant agreement n° 291763, and with the partial support of the DFG-funded Collaborative Research Centre 824: Imaging for Selection, Monitoring and Individualization of Cancer Therapies.

References

[1] 20q.net: The neural-net on the internet (www.20q.net).

[2] Akinator the web genie (http://en.akinator.com).

[3] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR 2010*, pages 3169–3176. IEEE, 2010.

[4] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *ICCV 2001*, volume 1, pages 105–112 vol.1, 2001.

[5] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 438–451. Springer Berlin Heidelberg, 2010.

[6] J. Carreira and C. Sminchisescu. CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[7] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(2):222–234, Feb 2014.

[8] A. Fathi, M. F. Balcan, X. Ren, and J. M. Rehg. Combining self training and active learning for video segmentation. In *Proceedings of the British Machine Vision Conference (BMVC 2011)*, volume 29, pages 78–1, 2011.

[9] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1–8. IEEE, 2009.

[10] L. Grady. Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1768–1783, Nov 2006.

[11] L. Grady, M.-P. Jolly, and A. Seitz. Segmentation from a box. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 367–374. IEEE, 2011.

[12] C. Gu, J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1030–1037, June 2009.

[13] A. Kowdle, D. Batra, W.-C. Chen, and T. Chen. imodel: interactive co-segmentation for object of interest 3d modeling. In *Trends and Topics in Computer Vision*, pages 211–224. Springer, 2012.

[14] A. Kowdle, Y.-J. Chang, A. Gallagher, and T. Chen. Active learning for piecewise planar 3d reconstruction. In *CVPR, 2011*, pages 929–936. IEEE, 2011.

[15] P. Krähenbühl and V. Koltun. Geodesic object proposals. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pages 725–739. Springer International Publishing, 2014.

[16] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, number MSR-TR-2009-85, 2009.

[17] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[18] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[19] C. Nastar, M. Mitschke, and C. Meilhac. Efficient query refinement for image retrieval. In *CVPR 1998*, pages 547–552, Jun 1998.

[20] B. Price, B. Morse, and S. Cohen. Geodesic graph cut for interactive image segmentation. In *CVPR 2010*, pages 3161–3168, June 2010.

[21] M. Sadeghi, G. Tien, G. Hamarneh, and M. S. Atkins. Hands-free interactive image segmentation using eyegaze. In *SPIE Medical Imaging*, pages 72601H–72601H. International Society for Optics and Photonics, 2009.

[22] T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. skr.*, 5:1–34, 1948.

[23] C. Straehle, U. Koethe, G. Knott, K. Briggman, W. Denk, and F. Hamprecht. Seeded watershed cut uncertainty estimators for guided interactive segmentation. In *CVPR, 2012*, pages 765–772, June 2012.

[24] K. Tieu and P. Viola. Boosting image retrieval. In *CVPR, 2000. Proceedings.*, volume 1, pages 228–235 vol.1, 2000.

[25] P. J. Toivanen. New geodosic distance transforms for gray-scale images. *Pattern Recognition Letters*, 17(5):437–450, 1996.

[26] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *ACM International Conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM.

[27] Z. Tu and S.-C. Zhu. Image segmentation by data-driven markov chain monte carlo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):657–673, May 2002.

[28] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2524–2531, Nov 2011.